100GE and 40GE PCS and MLD Proposal

IEEE 802.3ba    January   2008   Portland
Contributors and Supporters

David Law – 3com
Steve Trowbridge - Alcatel-Lucent
Brad Booth, Dimitrios Giannakopoulos, Francesco Caggioni, Keith Conroy – AMCC
Rita Horner – Avago
Arthur Marris – Cadence
Mike Shahine - Ciena
Mark Gustlin, Gary Nichol, Oded Trainin - Cisco Systems
Med Belhadj – Cortina
Chris Cole - Finisar
Krishnamurthy Subramanian – Force10
Aris Wong – Foundry Networks
Shinji Nishimura, Hidehiro Toyoda - Hitachi Ltd

Dan Dove – HP
Petar Pepeljugoski – IBM
Jerry Pepper, Thananya Baldwin - Ixia
Jeffery J. Maki, David Ofelt, Brad Turner - Juniper Networks
Martin White – Marvel
Pete Anslow – Nortel
Farhad Shafai - Sarance Technologies
Farzin Firoozmand, Craig Hornbuckle – SMI
Andre Szczepanek – TI
Martin Carroll - Verizon
Frank Chang - Vitesse
Agenda

• 40GE/100GE Architecture
• PCS and MLD layer details
• Possible XL/CGMII Interface
• Alignment details
• Alignment performance metrics
• Clocking example
• Skew
• Future work items and summary
MLD = Multi-Lane Distribution (AKA CTBI)
Proposed 100GE/40GE PCS and MLD Layer

• 10GBASE-R 64B/66B based PCS
  Run at 100G or 40G aggregate
• n Lane MAC/PCS to PMA/PMD Electrical Interface
  Ten Lanes for 100GE initially
  Four Lanes for 40GE initially
  Each lane runs at 10.3125G
  Data is striped across the electrical lanes 66 bit blocks at a time (round robin)
  Periodic alignment blocks are added to allow deskew
• Support m PMD lanes with the same PCS/MLD layer
• PMA maps n lane electrical interface to m lane PMD
  PMA is simple bit level muxing
  Does not know or care about PCS coding
• Alignment and skew compensation is done in the Rx MLD block only
This example is 40GE with 4 electrical and 4 optical lanes.

### PCS Functions:
- 66 bit encoding
- Scrambling

### MLD Functions:
- Alignment block addition periodically
- Round Robin block distribution

Each Block is a 66 bit Block

<table>
<thead>
<tr>
<th>TX PCS</th>
<th>TX MLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>
Alignment Mechanism – 40GE Example

RX MLD Functions:
- Re-Align 66 bit blocks
- Remove the Alignment blocks
Key Concept – Virtual Lanes

- This is only needed when the number of Electrical (n) and PMD (m) lanes are not equal.
  - If an interface will evolve so that n != m then VLs make the transition easy.

- Data from the MAC is first encoded into a continuous stream of 64B/66B blocks (100G or 40G aggregate stream).

- The 100G aggregate stream is split into a number of ‘virtual lanes’, also based on 64B/66B blocks.

- An alignment block is added to each virtual lane.
  - Sent infrequently.

- The number of virtual lanes generated is scaled to the Least Common Multiple (LCM) of the n lane electrical interface and the m lane PMD.
  - This allows all data (bits) from one virtual lane to be transmitted over the same electrical and optical lane combination.
    - This ensures that the data from a virtual lane is always received with the correct bit order at the Rx MLD.

- The virtual lane marking allows the Rx MLD to perform skew compensation, realign all the virtual lanes, and reassemble a single 100G or 40G aggregate stream (with all the 64B/66B blocks in the correct order).
Bit Flow Through – 100GE 4 lane PMD

- 20 VLs
- 10 Electrical lanes
- 4 Optical lanes
- With Skew, VLs move around
- RX MLD puts things back in order
How Many Virtual Lanes for 40GE?

- For each PMD objective, what is the number of lanes being considered?
  
  All can evolve to fewer lanes in the future

- Support at least 100m on MMF
  
  4 fibers

- Support at least 10m over a copper cable assembly
  
  4 lanes

- Support at least 1m over a backplane
  
  4 lanes

- Number of “Virtual Lanes” = Number of Lanes
  
  Just simple 66 bit block striping

<table>
<thead>
<tr>
<th>Number of Electrical Lanes</th>
<th>Supportable PMDs</th>
<th>Virtual Lanes Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>4, 2, 1</td>
<td>1, 2, 4</td>
<td>4</td>
</tr>
</tbody>
</table>

- With 4 VLs, all combinations of the above are possible
How Many Virtual Lanes for 100GE?

• For each PMD objective, what is the number of lanes being considered?
  
  All can evolve to less lanes in the future

• Support at least 10km on SMF
  
  4 wavelengths

• Support at least 100 meters on OM3 MMF
  
  10 fibers

• Support at least 40-km on SMF
  
  4 wavelengths

• Support at least 10m over a copper cable assembly
  
  10 lanes
How Many Virtual Lanes are Needed for 100GE?

<table>
<thead>
<tr>
<th>Number of Electrical Lanes</th>
<th>Supportable PMD Lane Widths</th>
<th>Virtual Lanes Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, 5, 4, 2, 1</td>
<td>1, 2, 3, 4, 5, 6, 8, 10, 12</td>
<td>120</td>
</tr>
<tr>
<td>10, 5, 4, 2, 1</td>
<td>1, 2, 3, 4, 5, 10</td>
<td>60</td>
</tr>
<tr>
<td>10, 5, 4, 2, 1</td>
<td>1, 2, 4, 5, 10</td>
<td>20</td>
</tr>
<tr>
<td>10, 5, 2, 1</td>
<td>1, 2, 5, 10</td>
<td>10</td>
</tr>
</tbody>
</table>

• It seems that the sweet spot is still 20 VLs…
• Supports currently envisioned PMDs
• Supports Electrical lane evolution such as a 4x25G interface, serial etc..20 VLs are always used
• Same 10GBASE-R PCS (Clause 49), untouched, just running at 40Gbps or 100Gbps

• Same Control Block encoding, same scrambler

With 8B alignment we don’t use all of the block types
XL/CGMII Interface

- Leverage XGMII
- Make it scalable
- Proposal: nx64 bit interface, logical interface
- nxTXD<63:0>, nxTXC<7:0>, nxRXD<63:0>, nxRXC<7:0>
- First 40GE might be 128bits + control
- First 100GE might be 320bits + control
- Only start packets on 8B boundary
  Simplifies MAC design, idle deletion etc.
- Use deficit counter to adjust idle to an average of 12B
Alignment Proposal

- Send alignment on a fixed time basis
- Alignment word also identifies virtual lanes
- Sent every 16384 66bit blocks on each virtual lane at the same time
  - \(\sim 216\mu\text{sec for 20 VLs @ 100G}\)
  - \(\sim 108\mu\text{sec for 4 VLs @ 40G}\)
- It interrupts packets
- Takes only 0.006% (60PPM) of the Bandwidth
- Rate Adjust FIFO will delete enough IPG so that the MAC still runs at 100.000G or 40.000G with the interface running at 10.3125G
Alignment Word Proposal

Requirements:
- Significant transitions and DC balanced – word is not scrambled
- Keep in 66 bit form, but no relation to 10GBASER is needed
- But why not keep it close? – Because of the clock wander concerns
- Contains Virtual Lane Identifier

Proposed Alignment Word

| 10 | VL | ~VL |

- This is DC balanced
- Encoding still TBD, will make it look random
- No relationship to the normal 10GBASE-R blocks
- Added after and removed before 64/66 processing
- Alignment block is periodic, no hamming distance concerns with 64/66 block types
Finding VL Alignment

- After reception in the rx MLD, you have x VLs, each skewed and transposed
- First you find 66bit alignment on each VL
  
  Each VL is a stream of 66 bit blocks
  
  Same mechanism as 10GBASE-R (64 valid 2 bit frame codes in a row)
- Then you hunt for alignment on each VL
  
  Look for one of the 20 VL patterns repeated and inverted four times
- Alignment is declared on each VL after finding 2 consecutive non-errored alignment patterns in the expected locations (16k words apart)
- Out of alignment is declared on a VL after finding 4 consecutive errored frame patterns
- Once the alignment pattern is found on all VLs, then the VLs can be aligned
Alignment Performance Parameters – 100GE

- **Mean Time To Alignment (MTTA)**
  Mean time it takes to gain Alignment on a lane or virtual lane for a given BER
  Nominal time = 314usec

- **Mean Time To Loss of Alignment (MTTLA)**
  Mean time it takes to lose Alignment on a lane or virtual lane for a given BER

- **Probability of False Alignment (PFA) = 3 E-40**

- **Probability of Rejecting False Alignment (PRFA) = ~1**

- Also have 64/66 stats on the graph for comparison
  - **MTTS – Mean Time To Sync**
  - BER **MTTLS – With the 125usec BER window, what is the Mean Time To Lose Sync**
  - **MTTLS - Mean Time To Lose Sync**
Variable Skew Handling

• This is a concern only for PMDs where we multiplex data when \( n \neq m \)

• We need to Find the maximum variable skew for the applicable PMDs

• For these PMDs, retiming buffers are used to handle the variable skew in the TX PMA and RX PMA

• Value of the variable skew is dependent on the technology, from a few bits to 10s of bits
Total Static Skew Numbers

• We need to add up all of the skew to see how much total static skew must be compensated for at the receiver
  • TX Electrical
  • TX PMD
  • Optical Medium
  • RX PMD
  • RX Electrical

• Note that 10nsec at 100Gbps = 1kbit of memory
• Numbers depend on what technology is ultimately used!
Work Items

• Determine Total Skew budget
• Determine Variable Skew budget
• Complete the MTTFPA
  Need likelihood of burst errors, electrical and optical
• State Machines
Summary

• Simple 10GBASE-R based PCS
• MLD layer to support multiple physical lanes/lambda
• Complexity is low within the MLD layer
  • Simple block data striping
• Complexity in the optical module is low
  • Simple bit muxing when m ≠ n
• Based on proven 64B/66B framing and scrambling
• Electrical interface is feasible at 10x10G or 4x10G
• Allows for a MAC rate of 100.000G or 40.000G
  • Overhead very low and independent of packet size
• Supports an evolution of optics and electrical interfaces